Hughes El Segundo Employees Association
Atari Computer Enthusiasts
Reprints of Exchange Newsletter Articles
October 1989 Richard L. Reaser

	Title of Article	Author's Last Name	Author's First Name	Newsletter	Issue Date	Remarks
2	2.5 MB Socketed RAM Upgrade for 1040ST	Orlando	Barry	PSAN	09/00/89	Replaces 256K chips with 1M chips.
5	Copy Protected Disk, Restoring	Golden	Jeff	DAL-ACE	07/00/89	Copy just good sectors from good disk to bad disk.
6	Target Practice	Thorson	Dave	NWPAC	09/00/89	Light gun program.
10	Trig Functions - Sine & Cosine, Using	Hogan	Brian	PSAN	09/00/89	Routines and explanations.
11	Upgrades the 130XE, Atari	Editor	Editor	SNACC	10/00/89	Mem tests the extra 64K, 48K, Keyboard Test, new PIA, etc

A 2.5 Meg Socketed Ram Upgrade for the 1040ST!

A Shareware Project by Barry Orlando (Dated: May 20, 1989)

(Copyright 1989 by Barry Orlando)

INTRODUCTION

This shareware text file *(downloaded from GEnie)* provides a procedure which details a 2.5 megabyte on board RAM upgrade project. The method described replaces 16 256K dynamic RAM chips with 16 one megabit dynamic RAM chips installed in chip sockets. Reinstallation of the ST's metal shield cover is also not impaired.

The beauty of this modification is that pin compatibility is simplified by the use of sockets instead of directly soldering the 1 meg RAM chips (DRAMs) to the motherboard and by the fact that the removed bank of sixteen perfectly reusable 256K DRAMs can be resold by you to persons upgrading 520STfm computers to 1 megabyte.

If you use this upgrade method, I recommend that you also utilize the high quality low contact resistance sockets that I've specified below (and not use any other standard sockets). These are of the same type and quality which are used on expansion boards made for the IBM clones.

I originally wrote this procedure to aid me in upgrading my own 1040ST's memory to 2.5 megabytes because I didn't trust myself to not make mistakes (and possibly damage any of the electronic components, especially the new DRAMs which I didn't consider cheap). I might note that my 1040ST's pc-board was marked Rev 4. But I don't believe that any other recent board revisions (if they exist) for boards with 32 256K DRAMs installed should affect this procedure.

Disclaimer of Liability...

I make no claim that this modification will work for you. I can only say that it worked for me. This modification should not be attempted by anyone except by someone with experience repairing or building digital electronic circuits. Performing this modification will be done at your own risk and may void the warranty on your computer.

BACKGROUND

The 256K dynamic ram chip differs from the 1 meg dynamic ram chip by having one additional connection, namely the address input A9. The following chip diagram illustrates this and shows pin layouts:

256K

A8	1.	16	VSS
Din	2	*15	CAS
WE	3	14	Dout
RAS	4	*13	A6
A0	5*	*12	A3
A2	6*	*11	A4
A1	7*	*10	A5
VCC	8*	*9	A7

1 Meg

Din	1.	**18	VSS
WE	2	17	Dout
RAS	3	*16	CAS
NC	4	15	A9
A0	5	*14	A8
A1	6*	*13	A7
A2	7*	*12	A6
A3	8*	*11	A5
VCC	9*	*10	A4

Truth Table for above chips:

A0-A9	Address Inputs
CAS	Column Address Strobe
Din	Data In
Dout	Data Out
RAS	Row-Address Strobe
VCC	Power (+5V)
VSS	Ground
WE	Read/Write Input
NC	Not Connected
*	pc-board pin compatible
**	pc-board pin compatible on most chips

The 2.5 megabyte modification essentially reconnects all pins as previously connected on the lower of two banks of RAM chips plus adds a new line from the previously unused A9 pin on the ST's MMU (pin 64 of U15) to all new 1 meg chip pins A9 via a new 33 ohm resistor. Some of the Address leads are interchanged, but this has no effect on the operation of the computer.

MAJOR TOOLS

1. Desoldering iron (I recommend the spring loaded type that incorporates the heat source.)

2. 15W grounded soldering iron (Radio Shack 64-2051 or similar)

3. Wire wrap clip and stripping tool

4. Exacto knife

5. 3-5 power magnifying glass or jewelers eyepiece

PARTS REQUIRED

1. One 33 ohm resistor, 1/4 watt, 5%.

2. 5 feet length of 30 gauge wire, Kynar wire wrap solid conductor *(Radio Shack 278-502)*.

3. 16 double contact low profile dual-in-line I.C. sockets *(D.C. Electronics catalog no. T02-18, cost: $0.12 each)*

4. Pack of 10 "Socket Wrap ID" *(D.C. Electronics catalog no. 16-ID, cost: $2.05). You can get in touch with D.C. Electronics at P.O. Box 3203, Scottsdale, AZ 85257, (800) 423-0070.*

5. 16 one megabit dynamic RAM chips, 120ns *(I used Hitachi Part no. HM511000-12, and paid $24.00 each plus state tax and $4.00 shipping from Ed at Computronix on Dec 16, 1988). You can get in touch with Ed at Computronix at 2151 A O'Toole Ave., San Jose, CA 95131, (408) 943-0771.*

WORK AREA

Prepare your work area by placing a towel on your flat working surface and cover it with a layer of aluminum wrapping foil. This provides padding for the components and a conductor for static electricity.

Precautions...

Since the slightest static discharge could easily damage components on your computer's pc-board, always keep one hand on the metal foil bonded along the outer rim of the pc-board before and during touching any part of the pc-board or its components. Avoid working on this project when the humidity is low.

PROCEDURE

1. After removing the pc-board from the insides of your computer, desolder and remove the front 16 256K chips (U33 through U45 and U16 through U30).

2. Using the Exacto knife, cut the following traces on the top side of the pc-board:

a. Between U30's pin 1 and U61's pin 1.

b. Between pin 14's of U33 through U45 and U16 through U30 and other pin 14's of U46 through U61.

c. Between the thru-hole (nearest to pin 1 in center of U32) of U32 and U46's pin 1. *(Note: This thru-hole traces to U32's pin 1 on the underside of the board.)*

3. Using the Exacto knife, cut the following traces on the bottom side of the pc-board:

a. Between U30's pin 1 and the 33 ohm resistor R76.

b. Between U32's pins 2 and 14. Do the same for U33 through U45 and U16 through U30.

c. Between U30's pin 4 and U15's pin 8. *(Note: Cut the trace after it goes through the board to the bottom side near resistor R91, but before it reappears on the top of the board via another thru-hole.)*

d. Between U32's pin 14 and the thru-hole located between U32 and U33.

4. Preparing New Sockets and Soldering Them Inplace:

a. On all 16 new sockets, remove the internals of pin 4.

b. On all 16 new sockets, bend all pins 1 and 17 flat against their bottom then up against their outer side.

c. On 14 of the new sockets, bend pin 18's flat against their bottom and pointing straight out to the side.

d. On the remaining two, bend both pin 18's flat against their bottom but back and down so it will just reach and fit into the pc-board hole of it's adjacent pin (pin 17). These two sockets will be installed and soldered with the socket's pin 18 going into the old 256K chip's pin 16 hole below the new socket's pin 17. Trial fit these two sockets at this time.

e. Using a pair of scissors, cut pieces of "Socket Wrap-ID" plastic markers for sandwiching them between the new sockets and the pc-board during installation. These little plastic markers should be cut so as to electrically isolate the socket's pins 1 and 17 from leads and traces on the pc-board, and from pin 18 of the socket for the two separately prepared sockets above, but also so as not to interfere with seating the socket's four pads flush with the top of the pc-board upon installation.

f. Solder the two sockets with the bent back pin 18 and markers into U30 and U33.

g. Solder the remaining sockets with markers into U34 through U45 and U16 through U29. Pin 18 of these socket's must be soldered to the top of the board at the nearest thru-hole that the pin 18 just barely reaches. Use plenty of heat as these thru-holes are big heat sinks.

5. Mounting The New 33 Ohm Resistor:

Install a new 33 ohm resistor on the pc-board by drilling two small holes from the top at points near capacitor C55 and resistor R70. Make sure you don't drill through a trace on the other side by holding the pc-board up to a bright light and marking the spots with a pencil.

6. Wiring:

Wire the following connections on the top side of the board:

a. Between U32's new socket's pins 1 and 17 and a thru-hole on the pc-board between U32 and U33. This thru hole is the only one that has a trace that runs to another thru-hole between U46 and U47 on the top side of the board.

b. Between U16's new socket's pins 1 and 17, and pin 2 of U54. Solder the wire directly to the side of the chip's pin 2.

c. Between each of the remaining new socket's pin 1 and 17 and a thru-hole in back of each 256K RAM chip (i.e., U47 through U53 and U55 through U61). These thru-holes are the ones that you will find traces back to either pin 2 or pin 14 of U47 through U61 and then trace over toward U22, U23, U26, and U27. Actually, pins 2 and 14 are tied together on the bottom of the board for every 256K RAM chip still installed. So that's 14 more wires altogether (i.e., each new RAM chip's pins 1 and 17 will be connected to the pins 2 and 14 of the 256K chip directly in back of it.

Wire the following connections on the bottom side of the board *(Note: all pin numbers here refer to the pin numbers on the pc-board, not the pin numbers of the new sockets or new ram chips)*:

d. Between U15's pin 64 and one side of the new 33 ohm resistor.

e. Between all pin 14's of U32 through U45 and U16 through U30, and the other side of the new 33 ohm resistor.

f. Between all pin 2's of U32 through U45 and U16 through U30, and the thru-hole near R91 which had its trace cut to U30's pin 4. This thru-hole traces back on top of the pc-board under the bank of resistors and over to U15.

g. Between U61's pin 1, U30's pin 4, and the cut trace side of resistor R76.

h. Between U30's pin 1 and U61's pin 3.

7. Inspect all solder joints with the magnifier for solder shorts and faulty connections. Verify that all connections were made correct.

8. Install the new 1 meg RAM chips into their new sockets.

9. That's it! Clean isn't it? Now sit back and admire your work.

10. Reassemble the computer and reconnect the monitor and power cord and let it whirl!

Shareware Policy

I am distributing this documentation as a $15.00 Shareware File. Please feel free to re-post on your favorite Information Service or local BBS. I ask only that you do so in its original unaltered form.

If you have followed these instructions and everything worked for you, I'm sure you will appreciate the effort I put into documenting this project and the money it has saved you from buying one of those expansion boards on the market or by the time you could have spent trying to figure such a project out by yourself. Please send your shareware user registration fee to:

Barry Orlando
1120 Deerfield Dr.
Napa, CA 94558

All registered users will be sent free of charge, a utility disk and color photographic prints of the completed project. The utility disk includes a program to test the new RAM and instructions on how to further modify the computer to a full 4 megabytes. The Utility Disk contains the following support files for the socketed 2.5 meg 1040ST modification:

1. SYSTAT7.PRG - A handy public domain utility which shows some neat things going on inside of your computer. Of special interest is the size of the ram.

(Continued on Page 23...)

```
  TRAP #NONE_LEFT
  OUT$(LEN(OUT$)+1)=IN$(A+F)
  # NONE_LEFT
  IN$=OUT$
  EXEC REPLACE
  # LINE_DONE
ENDPROC
---------------------------------
PROC GLITCH
  CLOSE
  POKE 559,34:POKE 566,146
  IF ERR<>136
     ? ,CHR$(253);"Error ";ERR
     ? ,"at line ";ERL
     END
  ENDIF
ENDPROC
---------------------------------
# OPEN_FILE_ERR
IN$=FILEIN$
IF M=8
  IN$=FILEOUT$
ENDIF
IF INSTR(IN$,":")=0
  IN$="Device ?:"
ENDIF
? "      Unable to open ";IN$
? CHR$(253)
? "   Check ";IN$(1,INSTR(IN$,":"));
 " and re-RUN"
---------------------------------
END :REM Unsucessful program exit
---------------------------------
*********************************
```


Please don't let your favorite computer get the kiss of death....

Help Stop Piracy!

CLASSIFIED ADS

HARDWARE AND SOFTWARE FOR SALE:

- Hard Drives & Accessories - Seagate 10 Meg HD (model ST412) - $49. Miniscribe 10 Meg HD (model 2012) - $49. IBM 5161 HD case (includes 130 watt power supply and expansion board) - $59. Or buy everthing for just $150!
- Atari SF-354 SSDD floppy disk drive, like new (still in box) - $59.
- ST Software: Championship Wrestling, Shuffle Board, Home Casino Poker, and Deep Space - $5 each or trade.

If interested or for more information contact Don Schenk at (206) 841-2820 (Puyallup, WA).

WANTED TO BUY: SYNFILE+ with documentation for the Atari XL/XE. Please call Marlene Bennett (before 9AM or leave message on machine) at (206) 833-4364 (Auburn, WA).

WANTED: Genealogy software for the Atari 8-Bit computers. Contact Ben Melton at (206) 851-6361 (Gig Harbor).

FOR SALE:

- ST Hard Drives: 97 Meg hard drive, 1 ST 296N (65 Meg), 1 Kayloc (30 Meg) with case, power supply and host adapter.
- Laser Printer: Brother HL8 with 1 Meg of memory. Fully HP compatible.
- Death Sword (ST game) - $10.

For more information and hardware prices call John Knierim at (206) 241-1175 (Seattle, WA)

WANTED TO BUY: 'Little Computer People' for the ST. Please contact Penny Ormston at (213) 436-5301 or via mail: 2291 West Williams Street, Apt #1, Long Beach, CA 90810.

BBS CALLERS WANTED: Try out 'Mr. Scooter's Domain' (an ANSI BBS) - (206) 824-3356 (Seattle, WA).

Attention all PSAN subscribers: *Get fast results when you want to buy, sell or trade Atari computer related items by placing an ad in this column. There is no charge for this service. Simply send your copy to PSAN in accordance with the Article Contributions instructions on page 1 or telephone directly to the PSAN Coordinator at (206) 566-1703 (Tacoma, WA).*

A 2.5 Meg Socketed Ram Upgrade for the 1040ST! (continued...)

2. BANKSTAT.TOS - A program that displays a table of possible memory configurations which are consistent with those published in the hardware specification from Atari Corp. (Developer's docs).

The program also reads into memory and displays your computer's current memory configuration for reference.

3. BANKSTAT.C - The C source code for BANKSTAT.TOS.

4. MEMTESTR.TOS - A program that performs a dynamic test of Bank 0 (the new memory bank of 1 megabit chips installed by 25MGUPGD.DOC) for data integraty and reports any defective DRAMs including the specific IC number.

5. MEMTESTR.C - The C source code for MEMTESTR.TOS.

6. 4MGUPGD.TXT - A text file that provides instructions for upgrading the 2.5 meg 1040ST to 4 megs.

The two 8x10 photographic color prints show the completed 2.5 meg modification on my 1040ST (one print for each side of the motherboard). These prints may be helpful since they show how the completed wiring should look.

Good Luck!

About the Author: *Barry Orland employed as an engineer with the Mare Island Naval Shipyard in Vallejo, CA. He is currently designing and building a custom 32,000 color board for the ST.*

****

5

RESTORING COPY PROTECTED DISKS

by Jeff Golden

We have had considerable success in restoring crashed floppy disks. What you need are a sector editor and a friend with the same version of the program. (Sometimes, if you have more than one drive, you may be able to find a drive that can read the bad disk, eliminating the need for the friend's copy.)

Since the bad disk is essentially worthless, you have little to lose by following these procedures. The worst you can do is make the disk worse than worthless, if there is such a thing.

1. Use the sector editor to determine the bad sectors on the defective disk, and write down the sector numbers. A sector copier may be able to perform this portion of the task much faster.

2. Use the same procedure on your friend's disk after making sure that it is write protected. Your friend's disk will probably show some bad sectors as part of the copy protection scheme. Do not attempt to copy these sectors.

3. Compare the sector before each bad sector to see if you have the same version of the program on both disks. Then copy only the sectors that are bad on your disk, but good on your friend's disk.

Presto, your worthless disk may run again. Next time, keep it away from paper clips and other metal objects, and don't leave it in the disk-drive when you turn the drive on or off. The next flip of the switch may be the unlucky one.

THE TROUBLE WITH FORMAT11

by Chris Freemesser

Reprint from ACORN, June 1989

So you have your ST, but you are unhappy with only 720k per DS disk. You find a format program that will increase your disk capacity to over 800k. You think to yourself "Gee, this is GREAT!". But is it? Maybe not.

A program called Format11 by Matt Orsie has been floating around for some time. I used this program on all my disks, be them SS or DS. Then I started having problems with my disks. Directories would not change when new disks were put in, garbled directories would pop up, and sometimes the computer crashed. At first, I thought my drive was going bad.

It was not the drive. It was the program! I found out that other people had the same problem, and the culprit was a logic error. Put simply, the ST assigns an identification number to each disk when it is formatted. The ST uses this (in part) to determine when a new disk is inserted. Format11 put the same number on EACH disk it formatted. The computer would get confused when it saw the same number on a new disk. I ended up spending an entire weekend copying my disks over to ones with a GOOD format on it.

So the warning is: DON'T USE FORMAT11! If you want to get more capacity out of your disks (and there is nothing wrong with this), I recommend DCFormat from Double Click Software. It is a SHAREWARE program, but is well worth it.

CHIPS/POWER

by Chip Scoppa

Reprinted from ACORN, June 1989

This column will appear from time to time in the Kernel and will deal with the use of LDW POWER (Logical Design Works' spreadsheet program).

First a little about Chip. I am an engineer and use PC's commonly in my everyday work. Up until a few months ago I knew very little about ATARI and even less about the ST, but I was fortunate enough to meet a few people who were committed to the ATARI way of computing.

After a couple of days at the new job here in the Rochester area I had scared up enough parts to put a PC on my desk without having to ask for the company to purchase a new one. I was pleased with the system that I had working and with the spreadsheet program that was on its hard drive.

The program was one that I was very familiar with and have enjoyed using. It is called LOTUS 1-2-3. (Now you can tell that the system which I was able to find was at least Big Blue compatible.)

Near to my office there were some unusual computers working and I got very interested when I started to see the kind of work that was being done using this new and strange kind of "PC". The work seemed to be of a better quality and seemed to be easier to do. I became more and more familiar with this new kind of "PC" and soon decided that I would like to have one of my own.

6

XE XL

TARGET PRACTICE

by Dave Thorson

NW PAC 9-89

Type-In Program for XL/XE Systems
Requires Atari Light Gun, BASIC

If you enjoy games that use the Atari Light Gun as much as I do, then you've no doubt noticed that after you buy a few there aren't any more. I decided that if I could throw together a quick program demonstrating how to write such games, maybe some of the more enterprising among you 8-bitters would take the hint and write something really incredible for the rest of us....

My program displays a target, like you might find at a shooting range, and lets you blast 10 holes in it to get a high score. I have more plans for it, such as being able to move the target back 25, 50 or 100 yards via the joystick to add more challenge. Effects due to wind or loss in elevation due to longer distances could also be factored in. Screen flipping could be used to draw the target on a hidden screen while the final score from the previous round is displayed, or perhaps a "fresh" target could be copied from memory rather than redrawn each time.

This program could also be the basis of a shooting gallery game, or whatever you could imagine. Once I add a bunch of bells and whistles, though, it will be far too lengthy for these pages. One nice surprise: I found the Atari Light Gun (at least mine, anyway) to be reasonably accurate. They're just not aligned properly to the sights at the factory. So, my program displays an alignment screen at the start. Carefully shoot the gun at the cross-hairs, using whatever sight alignment you prefer, and all your future shots will go right where the sights point. A routine to do this should be included in ALL light gun software -- the games would be a bit more enjoyable if the sights could be used.

Light Gun BASICS

First, how does a light gun work? It doesn't "shoot" anything at the TV, because there aren't any sensors there to pick it up. The process is actually the reverse: the light gun senses a beam of light from the TV. TV pictures are not a continuous display of color, but are made from focusing an electron beam on each color dot on the screen surface. The beam moves across the picture and then turns off, moves down a line, and scans across the next line. This process repeats for each scan line on the screen, then the beam shuts off and moves to the upper left corner of the screen to start over. This all happens sixty times each second, and somehow corresponds to the Vertical Blank Interrupt (VBI) in your Atari. If you could slow it way down, you would see a bright point of light skim across the first scan line, then the second, third, fourth... all the way to the bottom of the screen. When the bright spot passes by where the gun is aimed, the computer determines how long it's been since the last VBI. Through some simple subtraction of times based on scan rates, it can determine what line the beam is on and (a bit less accurately) how far across the line the beam has travelled.

I turned to my well-worn copy of Mapping the Atari for help in writing this one. I found memory locations for the light gun coordinates, but no mention of how to read the trigger. Since the gun plugs into a joystick port, I simply monitored the PORT A memory locations while playing with the trigger. It turns out that PEEK(54016) becomes 255 when the trigger is pressed. The horizontal coordinate (X) is read with PEEK(565) and the vertical (Y) with PEEK(564). All these locations and values hold whether the gun is plugged in to joystick port 0 or 1.

Gun coordinates are continuously updated as the gun moves around the screen. I took a hint from the commercial games I have and flash the screen brighter when the trigger is squeezed, then get the coordinates before restoring the normal colors. The coordinates must be scaled to graphics screen coordinates. The gun coordinate system horizontal values go from some number like 67 up to 228, and then wrap around to 0,1,2,3,... near the right screen edge. Vertically the coordinates range from 16 to 111 (according to Mapping the Atari). The alignment routine doesn't care about all this, it simply determines how much to shift the alignment coordinates to make them come out to 80 (X) and 40 (Y), the center of the GRAPHICS 7 screen coordinates.

Whenever a gun coordinate is read after that, it is shifted by the same amount to give a graphics

7

coordinate. Fortunately, GRAPHICS 7 pixels are about the same size as each point registered by the gun. For other graphics modes, additional scaling would be needed. One additional bit of shifting is needed because screen pixels are taller than they are wide (the Aspect Ratio). A variable (ADJ=8/7) is used to scale X-coordinates by the appropriate amount to normalize them with Y values. Circles appear round this way instead of oval. Gun coordinates must use the opposite scaling factor (FIX=7/8) to take the plotted point you are aiming at and convert it back to a value consistent with the radius value needed for scoring.

Program Description

The program should be understandable except for a few quirks of my programming style. The first few lines define values for statement labels used in the program. If WAIT=60, then a GOSUB WAIT command will jump to the routine at line 60. This makes the program more readable but causes a problem if you need to renumber things. The third line contains an ON...GOTO statement that contains all the defined label numbers. If the program is renumbered it will renumber these as well. Since they are stored in the same order as their definitions in the preceding lines, it's a simple matter to match the values with the names and change them to match the new line numbers.

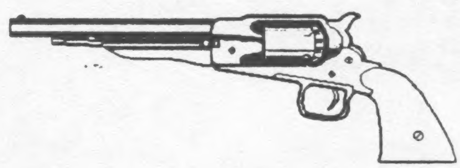
All REM's can be left out but they do make it easier to find your way around. There are no "funny" characters to type (you may want to use CTRL/T instead of * in the BULLET$ string). The listing is formatted as would appear on the screen (unless you change your margins).

For reasons of speed, all the time-critical subroutines are at the front of the program. The ON I GOTO statement at the front of the program is always forced to go to the INIT routine at the end of the program. INIT calculates the SIN and COS values needed by the CIRCLE routine in advance. It's much faster to get a value from an array than to compute sines and cosines during the game. The SIN values are also pre-scaled to compensate for the aspect ratio of the pixels described above. Since an accuracy of 15 degrees was acceptable, SIN and COS values are only computed for 15 degree increments even though the arrays SN and CS are dimensioned for 360 degrees in a trade-off of space for speed. If consecutive positions in SN and CS were used to store values at 15-degree increments, then degree values would need to be divided by 15 to find the result in the array; this would slow the calculations a bit more. The array RING() contains the radius for each scoring ring on the target, and VALUE() contains the value scored for each shot within the corresponding ring.

Sound values for the gun shot are stored in a series of arrays to track pitch, volume and distortion at each point in time. As the trigger is pulled, the screen is flashed and the sound started. The coordinates are determined, and then a loop in the NOISE routine plays through each of the remaining sound values. More complex than a simple SOUND command, but the sound is much nicer. Since the horizontal accuracy of the light gun is limited by timing considerations, the program reads the gun five times in a row and averages the values before scaling them to screen coordinates. This gives much better shooting accuracy and doesn't slow things too much. The GUNLOC routine also makes sure the resulting coordinates are within screen limits, plus some extra space for the width of the "holes" punched in the target.

The scoring section of the START routine takes a while since it must compute the radius from the target center to the shot location. This involves a slow square root computation, and then the score is looked up in the VALUE array, a bullet counter is subtracted from the BULLET$ string, and the results are displayed at the bottom of the screen. When the bullets are gone, a delay prevents you from shooting immediately; that would clear the score and start the next round. The text window is darkened during the delay period as a reminder.

Feel free to use any of the routines in your own programs, or to use the entire program as a starting point for something better. Simply replace the routines at REDRAW and START with your own. I would recommend avoiding anything involving circles because of all the required calculation overhead (ADJ and FIX to deal with the aspect ratio, plus sines, cosines and square roots, all of which s-l-o-w down a program). If you manage to create a masterpiece, please share it with the rest of us!



8

XE
XL

Target Listing

by Dave Thorson

```
10 INIT=660:WAIT=60:FLASH=100:NORMAL=1
30:HOLE=160:CIRCLE=190:NOISE=220:GUNLOC=250:S
HOOT=350:REDRAW=380
20 START=450:ALIGN=590:REM LABEL NAMES
30 I=1:ON I GOTO 660,60,100,130,160,19
0,220,250,350,380,450,590:REM LABEL VA LUES
40 REM
50 REM _______ WAIT _____ _________
60 POKE 540,W
70 ON PEEK(540)>0 GOTO 70 :RETURN
80 REM
90 REM              FLASH   ___________
100 POKE 708,12:POKE 709,12:POKE 710,1 2:POKE
711,12:POKE 712,12:RETURN
110 REM
120 REM __________ NORMAL _____________
130 POKE 708,44:POKE 709,4:POKE 710,15 4:POKE
711,15:POKE 712,154:RETURN
140 REM
150 REM              HOLE _____________
160 PLOT HPOS,VPOS:PLOT HPOS-1,VPOS-1: PLOT
HPOS-1,VPOS:PLOT HPOS,VPOS-1:PLOT
HPOS,VPOS:RETURN
170 REM
180 REM         ______ CIRCLE ______________
190 PLOT 80,40+R:FOR I=15 TO 360 STEP
15:DRAWTO 80+SN(I)*R,40+CS(I)*R:NEXT I :RETURN
200 REM
210 REM              NOISE _ _____ _ _____
220 W=1:FOR J=1 TO SMAX:SOUND 0,P(J),D
(J),V(J):GOSUB WAIT:NEXT J:RETURN
230 REM
240 REM _____ _____ GUNLOC ______________
250 W=1:GOSUB WAIT:GOSUB FLASH
260 VAVG=0:HAVG=0:FOR I=1 TO 5:VAVG=VA
VG+PEEK(VREG):HPOS=PEEK(HREG):HAVG=HAV
G+HPOS+228*(HPOS<50):NEXT I
270 GOSUB NORMAL:VPOS=INT(VAVG/5)
:HPOS =INT(HAVG/5):IF XC=0 THEN RETURN
280 HPOS=HPOS+XC:VPOS=VPOS+YC:IF HPOS< 2
THEN HPOS=2
290 IF HPOS>157 THEN HPOS=157
300 IF VPOS<2 THEN VPOS=2
310 IF VPOS>77 THEN VPOS=77
320 RETURN
330 REM
340 REM         ________ SHOOT _____________
350 SOUND 0,P(0),D(0),V(0):GOSUB GUNLO
C:RETURN
360 REM
370 REM ___________ REDRAW____________
380 COLOR 1:PLOT 125,79:DRAWTO 125,0:D RAWTO
35,0:POSITION 35,79:POKE 765,1:X IO 18,#6,0,0,"S:"
390 X0=80:Y0=40:R=RING(1):R2=R*R:COLOR 2
400 FOR Y=0 TO R:X=SQR(R2-Y*Y)*ADJ:PLO T
X0-X,Y0+Y:DRAWTO X0+X,Y0+Y:PLOT X0-X
,Y0-Y:DRAWTO X0+X,Y0-Y:NEXT Y
410 R=RING(3):GOSUB CIRCLE:R=RING(2):G OSUB
CIRCLE:COLOR 1:R=RING(0):GOSUB CI RCLE
420 RETURN
430 REM
440 REM ___________ START _____________
450 GRAPHICS 7:GOSUB NORMAL:GOSUB REDR
AW:POKE 752,1:? :?
460 BULLETS=10:B$="* * * * * * * * * *   ":SCORE=0:?
UP$;0,B$;"    000" 470 IF PEEK(GTRIG)<255 THEN 470
480 COLOR 0:GOSUB SHOOT:COLOR 0:GOSUB
HOLE:GOSUB NOISE
490 X=(HPOS-80)*FIX:Y=VPOS-40:R=INT(SQ
R(X*X+Y*Y)):V=0:IF R<40 THEN V=VALUE(R
):SCORE=SCORE+V
500 S$=STR$(SCORE+1000):BULLETS=BULLET
S-1:I=(10-BULLETS)*2:B$(I-1,I)="  "
510 IF BULLETS<1 THEN B$="  - Final Sc ore -   "
520 ? UP$;V,B$;"     ";S$(2)
530 ON PEEK(GTRIG)=255 GOTO 530:IF BUL LETS>0
THEN GOTO 470
540 POKE 710,152:W=150:GOSUB WAIT:GOSU B NORMAL
550 IF PEEK(GTRIG)<255 GOTO 550
560 GOTO START
570 REM
580 REM _____________ ALIGN ___________
590 GRAPHICS 7:GOSUB NORMAL:POKE 752,1 :? ," Gun
Alignment":? :? "    Point gun at center and fire";
600 COLOR 2:PLOT 0,40:DRAWTO 159,40:PL OT
80,0:DRAWTO 80,79
610 XC=0:YC=0
620 ON PEEK(GTRIG)<255 GOTO 620:GOSUB
SHOOT:GOSUB NOISE
630 XC=80-HPOS:YC=40-VPOS:RETURN
640 REM
650 REM _____________ INIT ____________
660 GRAPHICS 2+16:? #6:? #6:? #6;"  ta rget practice":?
#6:? #6;"  by dave th orson":? #6:? #6:? #6:? #6 670 ?
#6;"  PLEASE WAIT....":GTRIG=540
16:HREG=564:VREG=565:ADJ=8/7:FIX=7/8 680 DATA
2,8,14,5,8,12,11,8,10,20,8,8,
32,8,8,57,8,8,75,8,8,96,8,6,120,8,6,14
```

Bits & Pieces

9

by Joe Costa

```
7,8,6,170,8,6,185,8,6,200,8,6
690 DATA 215,8,4,230,8,4,243,8,4,250,8 ,4,253,8,2,0,0,0
700 SMAX=18:DIM P(SMAX),D(SMAX),V(SMAX
),UP$(1),B$(21),S$(4):UP$=CHR$(28)
710 DIM SN(360),CS(360):DEG :FOR I=0 TO 360 STEP
15:SN(I)=SIN(I)*ADJ:CS(I)=C OS(I):NEXT I
720 RESTORE 680:FOR J=0 TO SMAX:READ K
:P(J)=K:READ K:D(J)=K:READ K:V(J)=K:NEXT J
730 RESTORE 750: DIM VALUE(40),RING(4)
740 M=0:FOR I=0 TO 3:READ J,K:RING(I)= J:FOR M=M
TO J:VALUE(M)=K:NEXT M:NEXT  I:REM DATA IS
RADIUS & SCORE FOR RINGS
750 DATA 6,50
760 DATA 14,25
770 DATA 25,10
780 DATA 36,5
790 GOSUB ALIGN:GOTO START
```


Best Electronics
New Product Release

ST/MEGA Compatible Mouse

Features:

- Opto-Mechanical Design for Maximum user Sensitivity
- Ergonomical Design For Optimum user Comfort
- High Resolution Photo tracking of 200 Pulses per inch
- Replaceable Teflon wear pads (feet)
- ABS rollers on the steel photo optic interrupter shafts for maximum smooth operation
- FCC certified to comply within the limits for a Class B computing device
- Full ST/Mega owner support with replacement parts
- Compatible with 8 bit GEM operating systems
- Model CBM1 Mouse Suggested retail of $49.95

Now you have a second choice for mice you can use on your Atari ST/Mega* computer system! You will find because of its unique shape The Best Mouse can be used for manny hours of uninterrupted computing work with little or no fatigue. Most Atari owners who have used the Best Mouse agree, they would not switch back. See your local Atari dealer for your free TEST RIDE of this new premium mouse today or contact:

Best Electronics
2021 The Alameda Suite 290
San Jose, CA 95126
(408) 243-6950

*Atari, Atari ST, Atari Mega are registered Trademarks of Atari Corp

In case anyone hasn't noticed by now, Nybbles and Bytes has a new editor. I trust my first attempt in this issue won't cause me too much embarassment.

Steve Marshall has done a fantastic job and everyone owes him a tremendous vote of appreciation. He managed to find time in a very busy schedule to show me the rope. And now I find myself tied up in knots.

But he's not the only one. In assembling this issue and looking through back issues, I see certain names appearing time and again, such as Dave Thorson, Joe Krysa and Paul O. Parks. This newsletter will continue to welcome their contributions. But more of you out there have tales to tell and experiences to share and we need to hear from you.

Everyone has different tastes and interests. No one article will every appeal to everyone. In writing an article, that should not be a consideration. No one can afford to acquire every available piece of hardware or software for the Atari. Often, if we had the benefit of another's experience, we might make decisions on what to buy and what to use with a lot more confidence.

For those who would like some suggestions on possible articles, how about the following: reviews on productivity software such as word processing, spreadsheets, programming languages, utilities and telecommunications. And let's not forget games! There's a lot of public domain software and shareware out there that does fantastic things.

Some of us are hardware fanatics. There are many add ons and kits out there. Let's hear some success stories (and horror stories).

There are many sources of information about Atari computers in magazines and networks. Let's identify them and share our findings.

Cover Art --- a familiar symbol of our national pride by Dave Thorson.

PROGRAMMING FOR FUN

#9: Using Trig Functions - Sine and Cosine!

ST Tutorial by Brian Hogan, S*P*A*C*E

I thought I'd continue the spirit of the theme I started in my last article, namely that of highlighting a mathematical function. Of course I'll try to connect some graphics into the discussion. I'll look at the two basic trig functions, Sine and Cosine. Since trig is usually a semester course, I won't attempt to give all the details of how trig functions are used in ''real life'' in one short article!

The sine and cosine functions are basically the coordinates of the points on a unit (radius=1) circle as you move around the circumference. For this reason, the values of Sine and Cosine always lie between -1 and 1, inclusive. To see that connection with these functions, type in and run this first little program below. (The programs work best in LOW Resolution for these demos.)

```
FOR t=0 TO 6.2 STEP 0.05
   PLOT 30*COS(t)+50,30*SIN(t)+50
NEXT t
```

The values of t are marching along the circumference of the circle, the trig functions COS and SIN are giving the coordinates of the point on the circle at that particular spot along the circumference. The significance of the value of 6.2 is that it is appoximately equal to 2*pi, the circumference of the entire circle.

In plotting the circle on the computer screen, multiplying the values of SIN and COS by 30, magnifies the circle and the adding of 50 to the values moves the circle more to the middle of the screen. (You remember that the upper left corner is (0,0), right?)

The resulting figure won't quite look perfectly circular, due the the fact that the pixels on the screen and not perfectly ''square''. Play with different ''magnifications'' and placement of the circle.

Also, if you make the magnification factor different for the COS function than for the SIN function, you can get any elliptical shape you disire. Put in a DELAY or PAUSE in your loop in order to see the dots being drawn out.

Now, let's draw a graph of some SIN or COS ''waves''. See the following program:

```
DO
  COLOR (i MOD 14)+2 ! change color of
                       plotting.
  DEFTEXT (i MOD 14)+2 ! change color
                       of graphed text.
  TEXT 1,8*i+6,"Enter parameters a,b,c "
  LOCATE 24,i+1 ! LOCATE column, line --
                       Error in manual!
  INPUT "",a,b,c
  FOR x=0 TO 320
    PLOT x,a*SIN(b*x+c)+100
  NEXT x
  INC i
LOOP
```

This program will plot an x-y graph of the sine function. The parameters a,b,c will give you different variations of the sine graph. The value of a will determine the ''amplitude'', b will determine the ''frequency or period'', and b and c together determine the ''phase-shift''. If you picked the values of 1 for a and 1 for b and 0 for c, you would get just a little ''squiggle'' on the screen (try it!) That is mainly because the normal values of SIN are very small (-1 to 1) and that the sine function goes through one wave in just about 6 pixels (0 to 2*pi). To ''magnify'' the curve, put in values like 30 for a, and .1 for b (0 for c to begin with). Experiment, changing only one of the parameters, so you get a feel of the effect of that parameter. I made the program change color each time through the loop so you can keep track of the curve vs. the corresponding input. Adding the value of 100 to the y coordinate of the PLOT statement, moved the graph to the middle of the screen.

Now to finish off this month's programs, I'll add in a ''for pretty'' graphing program using the sine and cosine functions:

```
DO
   INPUT "step, offset (Try 1,56) ",stp,offset
   CLS
   FOR x=0 TO 319 STEP stp
      y1=30*SIN(0.1*x)
      y2=40*COS(0.15*(x+offset))
      COLOR (x/15 MOD 14)+2
      LINE x,y1+70,x+offset,y2+130
      PAUSE 0.1
   NEXT x
LOOP
```

The program figures out two y-values for two different trig functions. The program simply connects a line between those values. If you would imagine a trig curve traced out, then below it another trig curve, then pick a point on the top curve for a particular value of x, then for that same x add on an offset value and choose the coorresponding point on the lower curve. Then connect those two points by a line. Then move down to the next value of x and do the same thing. Of course play with different functions for the y1 and y2, choose a different scheme for choosing colors, etc!!!

Next time, I'll play with the polar coordinate system and the graphs of equations in polar coordinates. This will be a natural follow-up of this article, and we can get some ''beautiful'' pictures.

To ''toot'' my own horn a second, I just placed in Butler's Computer Service (located in Federal Way, WA) some copies of a small book I wrote for an elementary Math and Graphics class at Highline Community College. I just recently reworked the book so the demo programs are written in GFA BASIC. It is intended to explain the basics of computer graphics and the mathematics behind it. It does attempt to explain and teach the relevant mathematics needed to understand the details. Basic algebra is technically all that is needed if you're a ''sharp cookie''. The 125 page book comes with a disk that has all the demo programs. I've included on the disk a subroutine that can be added to your GFA programs to allow interactive input of functions into your programs. (No explanations of that subroutine, but source code is included!) All for the price of $19.95!! (Should've been a car salesman!) If you aren't in the area, just send me a letter (and $19.95) to

Brian Hogan
28802 57th Pl. S.
Auburn, WA 98001

and I'll send you the book postage free. (IN USA ONLY -- I have no idea of postage to faraway places.)

****

FUNCTION KEY LABELER

By Sid Kinne
16 bit VP

FunctionZ, $24.95 from Regent Software, is a new twist to an old idea. An innovative way of labeling the user defined function keys of your ST. The kit is a hardware/software package containing 6 plastic label stands, enough for 30 keys, and a program disk. No printed documentation is included although a short READ.ME file may be found on the program disk.

FunctionZ with lots of drop down menus and help screens is very user friendly, so simple an 8 year old could use it. Simply type the labels you need, 11 chracters wide by 4 lines and print them out. When the printer stops you will have two rows of neatly printed label inserts. All that remains is to cut out the labels, preoutlined by the program and slide them into the sturdy, well designed, plastic stands. Installation of the stands is just as easy. Slip them between the indicated function keys and the computer case, no tools are necessary.

All in all FunctionZ is a great product, at a great price and well worth it. Additional stands are available for $13.95 for 8. Anyone using redefined function keys should not be without these labels.

For a first hand look at the FunctionZ label system. Come to the club meeting and see how easy to use they are. I'll be bringing my ST and the labels I use.

* * *

ATARI UPGRADES THE 130XE

Atari Corp. has made the following changes to the production model 130XE computer.

Changes to the O.S. include:

A) MEMORY TEST now tests the extra 64K (in 4 squares).

B) MEMORY TEST checks the first 48K over TWICE as fast as before!

C) KEYBOARD TEST the function keys, F1-F4 keys, are missing from the screen display, although the code that interprets them is probably there.

D) "COPYRIGHT 1985 ATARI" replaces "COPYRIGHT 1983 ATARI", when keyboard tests are completed.

E) The O.S. chip is now on a 27256 EPROM, with only half of it being used! The original, which was on a 16K x 8 ROM, 27128.

4 - 41464 (4464) RAM chips replace the 16 - 4164 chips originally used

A completely different PIA chip--a 68B21! Compared to the 6520/6520A used in all other Atari 8-bits.

The new Owner's Manual (Rev. D), is now paper-bound, compared to spiral-bound on the original.

At least Atari went to the trouble of updateing the machine. It will probably saves them money, by being more reliable with less chip.

* * *

ST DRIVE POWER INDICATORS

By Wesley Ferrell
Down loaded from Genie

Warning: THIS WILL VOID YOUR WARRANTY because you must open your drive(s)!

For approximately $1.30 and about 5 mins. of time, you can put in a POWER ON indicator on your SF314 & SF354.

<u>Parts Needed</u>

1 pack of 2 LED's Radio Shack Part #276-037

1 package of 470 ohm resistors RS #271-1317

3.) Next open up your disk drive. Unscrew the 4 outer screws of the drive. Twist up and off to remove the top cover.

4.) Next examine the top cover to determine where you want to place the LED. This is a matter of taste so I won't get into location. Just remember you want to put the thing in the front!

5.) Now mark the center of the area that you going to place the LED and drill a 9/64th hole at the marked location. This will put the LED slightly recessed in the hole. You can drill a 5/32th hole and allow the LED to stick out. Again, this is a matter of taste.

6.) Now carefully examine the drive mechanisim. The drive is split into 2 pieces. Look at 2nd section. The 4-wire cable is the one you want to look at. Look at where the wires end. They will enter the board and be labeled 1 through 4. Pins 1 and 2 are the pins you want.

NOTE: The drives I have are 2 different styles (small eject button on the right and large flat eject button in front) The wires from pins 1 and 2 were different colors on each drive (blue and white on one drive, red and white on the other) Just remember that you want pins 1 and 2!

7.) Take a piece of wire approximately 6" long strip both ends about a .25" solder one end to pin 1. Now cut the ends of the resistor again about a .25" and solder the other end of the wire to one end of the resistor.

IMPORTANT: Now take the unsoldered leg of the resistor and solder it to the + or ANODE side of the LED. Again make sure that pin 1 is soldered to the + or ANODE side of the LED!

8.) Now take another piece of wire, strip the ends and solder one end to pin 2. Now solder the other end to the other leg of the LED. THAT'S IT!

9.) Plug in the power supply and turn on the drive. You should get a soft green or red glow, if not TURN OFF THE DRIVE. Check for bad solder connections or a short across the pins 1 and 2 Correct as necessary.

(Continued on page 5)

(Continued from page 3)

10) Determine the opening size the fan will draw through. Use a hole cutter to cut the hole or drill a pattern of holes over this area.

11) Mount the fan to the RF shield. Be sure the fan blows upward. (It won't be forcing dust into your machine this way.)

11) Reinstall the RF shield, replace the screws, and give those twist tabs a turn.

12) Reinstall the power supply (don't forget the mounting screws) but don't reconnect to the motherboard. As this is where you will connect the fan. The connector has 5V, 12V and a (black) ground wire.

13) Plug in the power cable, turn on the power supply. BE CAREFUL! YOU ARE NOW LIVE! Use a volt-meter (if available) to determine which lines are which 5V or 12V. You could just connect the fan's wires to determine the proper connection, as a DC fan is polarity sensitive. If nothing happens, reverse the wires and try again. Make a note of which wires to connect and shut off the power.

14) Remove the 12V and ground wires from the connector by pressing on the small metal tabs with a screwdriver then pull them out. Cut the fan wires to length, strip back about 1/2" of the insulation, and soldered to the appropriate connector and snap back into the connector.

15) After connecting the fan, turn on the power supply and check that the fan is working properly. Listen for a vibrations and adjust as necessary.

16) If everything is OK, reasseab your computer in the reverse order disassembly: Reconnect the power supp to the motherboard, Reinstall di drive and cables, replace power supp and disk drive shields, replac keyboard and reconnect to motherboard

17) Now connect your monitor and boot disk to check you haven't forgott anything. If everything is still O replace the top cover, and case screw watch out for different screw length

18] The Fan installation is n completed! and You have an ATARI that will run cooler and should la longer. Especially if you're running BBS. Hope all who try th modification have no problems. If y take your time, follow t instructions, and use standa precautions regarding static etc, y should have no problems.

* * *

(Continued from page 4)

10.) Place the LED in the hole y drilled and secure with epoxy, supe glue or silicone gel. I used the g myself.

11.) Close cover on the drive and p the screws back in. Tighten them a enjoy!

* * *